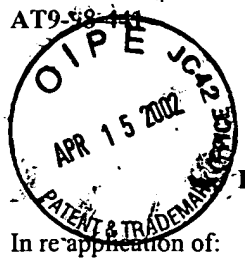
AT9-98-444

PATENT

AF 2700



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Chao et al.

Serial No.:	09/282,907	Art Unit:	2172
Filed:	March 31, 1999	Examiner:	J. Fleurantin
For:	ERROR DETECTION PROTOCOL		

Box AF
Assistant Commissioner for Patents
Washington, D.C. 20231

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION - 37 CFR 1.192)

1. Transmitted herewith in triplicate is the APPEAL BRIEF in this application with respect to the Notice of Appeal filed on March 5, 2002.

NOTE: "The appellant shall, within 2 months from the date of the notice of appeal under § 1.191 in an application, reissue application, or patent under reexamination, or within the time allowed for response to the action appealed from, if such time is later, file a brief in triplicate." 37 CFR 1.192(a) (emphasis added).

2. STATUS OF APPLICANT

This application is on behalf of

- ☒ other than a small entity
- ☐ small entity

  verified statement:
  - ☐ attached
  - ☐ already filed

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 CFR 1.17(f) the fee for filing the Appeal Brief is:

☐	small entity	$160.00
☒	other than a small entity	$320.00
	Appeal Brief fee due	**$320.00**

CERTIFICATE OF MAILING (37 CFR 1.8)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to Box AF, Assistant Commissioner for Patents, Washington, D.C. 20231.

Date: 4/12/02

Serena Beller
(Type or print name of person mailing paper)

Serena Beller
(Signature of person mailing paper)

4. EXTENSION OF TERM

NOTE: The time periods set forth in 37 CFR 1.192(a) are subject to the provision of § 1.136 for patent applications. 37 CFR 1.191(d). Also see Notice of November 5, 1985 (1060 O.G. 27).

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply.

(complete (a) or (b) as applicable)

(a) ☐ Applicants petition for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

Extension (months)	Fee for other than small entity	Fee for small entity
☐ one month	$ 110.00	$ 55.00
☐ two months	$ 400.00	$200.00
☐ three months	$ 920.00	$460.00
☐ four months	$1,960.00	$980.00
	Fee	$_____

If an additional extension of time is required, please consider this a petition therefor.

(check and complete the next item, if applicable)

☐ An extension for _____ months has already been secured and the fee paid therefor of $_____ is deducted from the total fee due for the total months of extension now requested.

Extension fee due with this request $_____

or

(b) ☒ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicants have inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE

The total fee due is:

Appeal Brief fee $320.00

Extension fee (if any) $0.00

TOTAL FEE DUE $310.00

6. FEE PAYMENT

☐ Attached is a check in the sum of $____

☒ Charge Account No. 09-0447 (AT9-98-441) the sum of $320.00.

A duplicate of this transmittal is attached.

7. FEE DEFICIENCY

NOTE: If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum, six-month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays are encountered in returning the papers to the PTO Finance Branch in order to apply these charges prior to action on the cases. Authorization to charge the deposit account for any fee deficiency should be checked. See the Notice of April 7, 1986, 1065 O.G. 31-33.

☒ If any additional extension and/or fee is required, this is a request therefor and to charge Account No. 09-0447 (AT9-98-441).

AND/OR

☒ If any additional fee for claims is required, charge Account No. 09-0447 (AT9-98-441).

Reg. No.: 47,159

SIGNATURE OF ATTORNEY

Tel. No.: (512) 370-2832

Robert A. Voigt, Jr.
WINSTEAD SECHREST & MINICK P.C.
5400 Renaissance Tower
1201 Elm Street
Dallas, Texas 75270

::ODMA\PCDOCS\AUSTIN_1\187598\1
1162:7047-P280US

#12 WM 4-30-02

AT9-98-441

PATENT

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Chao et al.	:	Before the Examiner: J. Fleurantin
Serial No.: 09/282,907	:	Group Art Unit: 2172
Filed: March 31, 1999	:	IBM Corporation
	:	Intellectual Property Law
Title: ERROR DETECTION PROTOCOL	:	11400 Burnet Road
	:	Austin, Texas 78758
	:	

April 12, 2002

APPEAL BRIEF

Box AF
Assistant Commissioner for Patents
Washington, D. C. 20231

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, which is the assignee of the entire right, title and interest in the above-identified patent application.

CERTIFICATION UNDER 37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Box AF, Assistant Commissioner for Patents, Washington, D.C. 20231, on April 12, 2002.

Serena Beller
Signature
Serena Beller
(Printed name of person certifying)

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-25 are pending in the Application. Claims 1-25 stand rejected.

IV. STATUS OF AMENDMENTS

The Appellants' response to the Office Action having a mailing date of October 22, 2001, has been considered, but the Examiner indicated that it did not place the application in condition for allowance because the Appellants' arguments were deemed unpersuasive.

V. SUMMARY OF INVENTION

The apparatus of the present invention is for providing a recent set of replicas for a cluster data resource. Specification, Page 6, Lines 8-9. The apparatus has a cluster having a plurality of nodes in a peer relationship, each node has an electronic memory for storing a local replica of the cluster data resource. Specification, Page 6, Lines 9-11. A group services client, which is executable by each node of the cluster, has cluster broadcasting and cluster voting capability. Specification, Page 6, Lines 11-12. A database conflict resolution protocol ("DCRP"), which is executable by each node of the cluster, interacts with the group services clients such that the DCRP broadcasts to the plurality of nodes a data resource modification request having a data resource identifier and a timestamp. Specification, Page 6, Lines 12-15. The DCRP determines a recent replica of the cluster data resource among the nodes with respect to the timestamp of the broadcast data resource modification request relative to a local timestamp associated with the data resource identifier, and distributes the

recent replica of the cluster data resource to each required node of the plurality of nodes. Specification, Page 6, Lines 15-19.

VI. ISSUE

Are claims 1-25 properly rejected under 35 U.S.C. §103(a) as being unpatentable over San Andres et al. (U.S. Patent No. 5,956,489) (hereinafter "San Andres")?

VII. GROUPING OF CLAIMS

Claims 9 and 15 form a first group.

Claims 10 and 16 form a second group.

Claims 11, 17 and 22 form a third group.

Claims 12, 18 and 23 form a fourth group.

Claims 13, 19 and 24 form a fifth group.

Claims 14, 20 and 25 form a sixth group.

Claims 1-8 and 21 should not be grouped together and should each be considered separately.

The reasons for these groupings are set forth in Appellants' arguments in Section VIII.

VIII. ARGUMENT

Claims 1-25 are not properly rejected under 35 U.S.C. §103(a) as being unpatentable over San Andres

A *prima facie* showing of obviousness requires the Examiner to establish, *inter alia*, that the prior art references teach or suggest, either alone or in combination, all of the limitations of the claimed invention, and the Examiner must provide a motivation or suggestion to combine or modify the prior art reference to make the claimed inventions. M.P.E.P. §2142. The motivation or suggestion to combine references must come from one

of three possible sources: the nature of the problem to be solved, the teachings of the prior art, and the knowledge of persons of ordinary skill in the art. *In re Rouffet*, 47 U.S.P.Q.2d 1453, 1458 (Fed. Cir. 1998). The showings must be clear and particular. *In re Dembiczak*, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). Broad conclusory statements regarding the teaching of multiple references, standing alone, are not evidence. *Id.*

In order to reject under 35 U.S.C. §103, therefore, the Examiner must provide a proper motivation for combining or modifying the references. M.P.E.P.§2142; *In re Rouffet*, 47 U.S.P.Q.2d 1453, 1457-1458 (Fed. Cir. 1998). The Examiner recites that "it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teaching of San Andres with the step of each node in the computer cluster voting *based on a functional outcome of the database update request. This modification would allow the teachings of San Andres to provide access to identical data and so that the on line service appears the same to all end users.*" Office Action (dated July 3, 2001), Pages 3-4. The Examiner further recites as motivation that "San Andres provides user access to service content data that is updated by the on line service on a transaction by transaction basis, and permits users to read and download messages for review by other users, see col 1, lines 38-43." Advisory Action, Page 2; Office Action (dated September 25, 2001), Page 2.

There is no motivation to modify San Andres as there is *no suggestion or motivation* in San Andres or in the knowledge of those ordinary skilled in the art *so that each application server votes based on a functional outcome of the database update request.* As stated above, the Examiner points to providing users access to service content data that is updated by the on line service on a transaction by transaction basis and permitting users to read and download messages for review by other users as motivation for modifying San Andres. However, the Examiner's motivation for modifying San Andres so that users are able to read and download messages does not explain or imply that San Andres should be

modified *so that each application server votes based on a functional outcome of the database update request.* The Examiner's only support for modification is his own subjective opinion, which is insufficient. The Examiner must submit *objective evidence* in support of modifying San Andres. *In re Lee*, 61 U.S.P.Q.2d 1430, 1433 (Fed. Cir. 2002); *In re Kotzab*, 55 U.S.P.Q.2d 1313, 1316-1317 (Fed. Cir. 2000). Furthermore, San Andres teaches that "*one problem with the two-phase commit protocol is that it is poorly suited for an on-line services network* that handles large numbers of concurrent user connections." Column 2, Lines 13-15. San Andres further teaches that "what is needed, therefore, is a *mechanism for efficiently processing update requests made to replicated, transaction-based services.*" Column 2, Lines 20-22. San Andres further teaches that "what is also needed is an *efficient mechanism for bringing the content of an application server up-to-date with that of other application servers, so that new application servers can be added to service group* ... and so *that existing application servers can efficiently be taken off-line for maintenance.*" Column 2, Lines 23-30. As interpreted by the Appellants, the *purpose of San Andres is to ensure that existing application servers can efficiently be taken off-line for maintenance* which was a problem with the two-phase commit protocol. There would be no motivation to provide users access to service content data that is updated by the on line service on a transaction by transaction basis and permitting users to read and download messages for review by other users in San Andres as the *purpose of San Andres is to ensure that existing application servers can efficiently be taken off-line for maintenance.* Therefore, there is no motivation to modify San Andres as there is *no suggestion or motivation* in San Andres or in the knowledge of those ordinary skilled in the art to modify San Andres *so that each application server votes based on a functional outcome of the database update request.*

As the Appellants have previously shown, there is no motivation to modify San Andres as the *proposed modification would render the invention in San Andres unsatisfactory for its intended purpose* as explained below and therefore there is no

suggestion or motivation to make the proposed modification. Reply under 37 C.F.R. §1.111, mailed July 19, 2001, pages 4-5; *In re Gordon*, 733 F.2d 900, 221 U.S.P.Q. 1125 (Fed. Cir. 1984); M.P.E.P. §2143.01. *The Examiner has not addressed this in any of the previous Office Actions. Where the Appellants traverse a rejection, the Examiner should answer the substance of Appellants' argument if the Examiner repeats the rejection. M.P.E.P. §707.07(f).* Furthermore, the *proposed modification would change the principle of operation of San Andres* as explained below and therefore the teachings of San Andres are not sufficient to render the claims *prima facie* obvious. Reply Under 37 C.F.R. §1.111, mailed July 19, 2001, pages 4-5; *In re Ratti*, 270 F.2d 810, 123 U.S.P.Q. 349 (C.C.P.A. 1959); M.P.E.P. §2143.01. *The Examiner has not addressed this in any of the previous Office Actions. Where the Appellants traverse a rejection, the Examiner should answer the substance of Appellants' argument if the Examiner repeats the rejection. M.P.E.P. §707.07(f).* San Andres teaches that "the servers 120 that receive the update transaction from the Arbiter respond by processing the update transaction, and by returning a status code that indicates the success or failure of the transaction." Column 19, Lines 25-28. San Andres further teaches that "the update transactions are *dispatched to the servers 120 in a serialized order ... so that all servers of the service group process the update transactions in the same order. This ensures consistency between the replicated copies of the service content data.* Each time an update transaction is dispatched by the Arbiter, the Arbiter monitors the outcome ('success' or 'failure') of the transaction on each server 120 by checking the status codes returned by the servers 120. *When one server 120 of the service group processes the dispatched transaction differently than the other servers, the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group.* In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent' with the final outcome, and are taken off-line." Column 19, Lines 37-55. As interpreted by the Appellants,

San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off line for maintenance.* In other words, the correct processing is determined to be that the majority have processed the transaction consistently and the minority have processed the transaction inconsistently.

San Andres further teaches that "what is also needed is an efficient mechanism for bringing the content of an application server up-to-date with that of other application servers *... so that existing application servers can efficiently be taken off-line for maintenance.*" Column 2, Lines 23-30. As interpreted by the Appellants, *the purpose of San Andres is to ensure that existing application servers can efficiently be taken off-line for maintenance. However, by having each application server implement a voting scheme to indicate whether there is a difference between the broadcast results and the local modification-result code; the Arbiter would not be able to decide which application servers are inconsistent and thus be taken off-line for maintenance.* Hence, the *proposed modification would render the invention in San Andres unsatisfactory for its intended purpose* and therefore there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 U.S.P.Q. 1125 (Fed. Cir. 1984); M.P.E.P. §2143.01. Furthermore, the *proposed modification would change the principle of operation of San Andres* and therefore the teachings of San Andres are not sufficient to render the claims *prima facie* obvious as a matter of law. *In re Ratti*, 270 F.2d 810, 123 U.S.P.Q. 349 (C.C.P.A. 1959); M.P.E.P. §2143.01.

The Examiner contends that "it must be recognized that any judgment on obviousness is necessarily a reconstruction based upon hindsight reasoning. *In re McLaughlin*, 433 F.2d 1392, 170 U.S.P.Q. 209 (C.C.P.A. 1971)." Advisory Action, Page 2; Office Action (dated September 25, 2001), Page 2. However, the Examiner's reliance on *In re McLaughlin* is

misplaced. *In re McLaughlin* does not relieve the Examiner of the *Examiner's burden of demonstrating by objective evidence that the obviousness rejection takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made and does not include knowledge gleaned from the Appellants' disclosure.* Indeed, it is now well settled that it must be demonstrated by objective factual evidence that the obviousness rejection does not include knowledge gleaned only from the Appellants disclosure, and, for example, relying on the Appellants disclosure to find prior art corollaries of the claim limitations is an improper hindsight reconstruction. *In re Lee*, 61 U.S.P.Q.2d 1430, 1433 (Fed. Cir. 2002); *In re Dembiczak* 175 F.3d 994, 999, 50 U.S.P.Q.2d 1613, 1617 (Fed. Cir. 1999); *In re Rouffet*, 149 F.3d 1350, 1357, 47 U.S.P.Q.2d 1453,1457 (Fed. Cir. 1998). Moreover, broad conclusory statements regarding the teachings of multiple references, standing alone, are not evidence. *In re Dembiczak,* 175 F.3d 994, 999, 50 U.S.P.Q.2d 1613, 1617; accord *In re Lee*, 61 U.S.P.Q.2d 1430, 1435 (Fed. Cir. 2002); *In re Kotzab,* 217 F.3d 1365, 1370, 55 U.S.P.Q.2d 1313, 1317 (Fed. Cir. 2000); M.P.E.P. § 2143.01 (explaining *In re Kotzab*).

San Andres does not teach or suggest "*each node* in the computer cluster *voting based on a functional outcome of the database update request*" as recited in claim 1. The Advisory Action and Office Action (dated September 25, 2001) state that "San Andres strongly suggests the step of the arbiter monitors the outcome of the transaction on each server by checking the status codes returned by the servers, when one server of the service group processes the dispatched transaction differently than the other servers the arbiter uses a voting scheme to decide which server or servers are to be taken off line service group, the arbiter uses a majority rules voting scheme under the majority rules scheme, if the majority number servers of the service group report a different outcome than others servers the majority servers are treated as being inconsistent with final outcome and taken off line; which is read as each node in the computer cluster voting based on a functional outcome of

the database update request (see col. 19, lines 44-55)." Advisory Action, Page 2; Office Action (dated September 25, 2001), Pages 2-3. Appellants respectfully contest the assertion that the above cited passage teaches or suggests that *each node* in the computer cluster *voting based on a functional outcome of the database update request.* As stated above, as interpreted by the Appellants, San Andres simply teaches an *Arbiter* that may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off line for maintenance.* Consequently, San Andres does not teach or suggest that *each node* in the computer cluster *voting based on a functional outcome of the database update request.* Furthermore, the Advisory Action and Office Action (dated September 25, 2001) state that "this modification would allow the teachings of San Andres to improve the accuracy and reliability of the error detection protocol." *Id.* The Examiner's only support for modification is his own subjective opinion, which is insufficient. The Examiner must submit *objective evidence* in support of modifying San Andres. *In re Lee,* 61 U.S.P.Q.2d 1430, 1433 (Fed. Cir. 2002); *In re Kotzab,* 55 U.S.P.Q.2d 1313, 1316-1317 (Fed. Cir. 2000). For at least the reasons stated above, Appellants respectfully assert that it would not be obvious to modify the teachings of San Andres with the step of each node in the computer cluster voting *based on a functional outcome of the database update request* and therefore the Examiner has not presented a *prima facie* case of obviousness. Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 1 in view of the cited prior art.

San Andres does not teach or suggest "*detecting* an *out-of-sync condition* as a *result of a different functional outcome*" as recited in claim 1. Instead, San Andres teaches that "*when one server 120 of the service group processes the dispatched transaction differently than the other servers, the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group.* In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, *if a minority number of*

servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent' with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance.* As interpreted by the Appellants, *San Andres simply teaches an Arbiter using a scheme to determine which servers are to be taken off-line but does not teach detecting an out-of-sync condition as a result of a different functional outcome.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 1 in view of the cited prior art.

San Andres does not teach or suggest that "the *out-of-sync condition is an error*" as recited in claim 2. Instead, San Andres teaches that "*when one server 120 of the service group processes the dispatched transaction differently than the other servers, the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group.* In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent'* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance.* As interpreted by the Appellants, *San Andres simply teaches an Arbiter using a scheme to determine which servers are to be taken off-line but does not teach detecting an out-of-sync condition that is an error.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 2 in view of the cited prior art.

San Andres does not teach or suggest that "*refreshing* the database *in response to the detecting step*" as recited in claim 3. Instead, San Andres teaches that "when an application server of a service group receives a client request that indicates a modification to replicated service content data, the server/service generates an update transaction and sends the update transaction to the Arbiter. The Arbiter records the update transaction in a service-group specific transaction log ... and *forwards the transaction for immediate processing to every application server* in the server group ... *The application servers process the update transaction, and return status codes indicating, for each respective application server, the 'success' or 'failure' of the transaction.*" Column 3, Lines 26-38. San Andres further teaches that "*when different application servers return different status codes, the Arbiter uses the ... conflict resolution feature to resolve the conflict.*" Column 3, Lines 41-44. San Andres further teaches that "when different application servers of a service group process the same update transaction differently, the Arbiter resolves the conflict by determining the 'final out-come' of the transaction for the service group as a whole, and by taking any application servers off-line that are in conflict with this final outcome." Column 2, Lines 55-60. As interpreted by the Appellants, San Andres teaches an arbiter that sends an update transaction to each application server to be processed. Upon the application servers processing the update transactions, the arbiter may be configured to resolve a conflict where the conflict being different status codes sent by different application servers upon processing the update transaction. As interpreted by the Appellants, the *application servers do not process the update transaction in response to detecting an out-of-sync condition*. The *application servers simply process the update transaction sent by the arbiter*. Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 3 in view of the cited prior art.

San Andres does not teach or suggest that "*resetting cluster membership in response to the detecting step*" as recited in claim 4. Instead, San Andres teaches that "when one

server 120 of the service group processes the dispatched transaction differently than the other servers, *the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group*. In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent'* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance.* As interpreted by the Appellants, *San Andres simply teaches an Arbiter using a scheme to determine which servers are to be taken off-line but does not teach resetting cluster membership in response to detecting an out-of-sync condition.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 4 in view of the cited prior art.

San Andres does not teach or suggest "*blocking* further *participation* by the *node having the out-of-sync condition in response to the detecting step*" as recited in claim 5. San Andres teaches that "when different application servers of a service group process the same update transaction differently, the Arbiter resolves the conflict by determining the 'final out-come' of the transaction for the service group as a whole, and by taking any application servers off-line that are in conflict with this final outcome." Column 2, Lines 55-60. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance.* San Andres does not teach *blocking participation* by the *node that has the out-of-sync condition* in response to detecting the out-of-sync condition. Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 5 in view of the cited prior art.

San Andres does not teach or suggest "declaring an end-of-transaction state on *update voting completion* when the database update is being done in a transactional manner" as recited in claim 6. Instead, San Andres teaches that "the servers that receive the update transaction from the Arbiter respond by processing the update transaction, and by returning a status code that *indicates the success or failure of the transaction.*" Column 19, Lines 25-28. As interpreted by the Appellants, San Andres teaches that the servers process the update transaction sent from the Arbiter and then return a status code that indicates whether the *server was successful or not on processing the update transaction.* San Andres does not teach that the servers declare an end-of-transaction state *on update voting completion* when the database update is being done in a transactional manner. Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 6 in view of the cited prior art..

San Andres does not teach or suggest "*backing out an update when update voting does not meet a criteria established for success*" as recited in claim 7. Instead, San Andres teaches that "the servers that receive the update transaction from the Arbiter respond by processing the update transaction, and by returning a status code that *indicates the success or failure of the transaction.*" Column 19, Lines 25-28. As interpreted by the Appellants, San Andres simply teaches that the servers process the update transaction sent from the Arbiter and then return a status code that indicates whether the *server was successful or not on processing the update transaction.* San Andres does not teach that the Arbiter or server application *backs out an update when the update voting does not meet a criteria established for success.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 7 in view of the cited prior art.

San Andres does not teach or suggest that "the criteria for success is that *no more than one node has inconsistent results*" as recited in claim 8. Instead, San Andres teaches that "when one server 120 of the service group processes the dispatched transaction

differently than the other servers, *the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group.* In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent'* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance.* San Andres *does not teach that the Arbiter's voting criteria is that no more than one node has inconsistent results.* Instead, as interpreted by the Appellants, the Arbiter's voting criteria is to treat the *minority number* of servers *with a different outcome than the majority number* of servers *as inconsistent.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 8 in view of the cited prior art.

San Andres does not teach or suggest "*voting, by all of the other nodes in the computer cluster*, to approve update if a match results from the comparison" as recited in claims 9 and 21 and similarly claim 15. The Advisory Action and Office Action (dated September 25, 2001) state that "San Andres strongly suggests the step of the arbiter monitors the outcome of the transaction on each server by checking the status codes returned by the servers, when one server of the service group processes the dispatched transaction differently than the other servers the arbiter uses a voting scheme to decide which server or servers are to be taken off line service group, the arbiter uses a majority rules voting scheme under the majority rules scheme, if the majority number servers of the service group report a different outcome than others servers the majority servers are treated as being inconsistent with final outcome and taken off line; which is read as each node in the computer cluster voting based on a functional outcome of the database update request (see col. 19, lines 44-55)." Advisory Action, Page 2; Office Action (dated September 25, 2001), Pages 2-3. Appellants

respectfully contest the assertion that the above cited passage teachs or suggests that *each node* in the computer cluster *voting based on a functional outcome of the database update request*. As stated above, as interpreted by the Appellants, San Andres simply teaches an *Arbiter* that may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off line for maintenance.* Consequently, San Andres does not teach or suggest that *each node* in the computer cluster *voting based on a functional outcome of the database update request.* Furthermore, the Advisory Action and Office Action (dated September 25, 2001) state that "this modification would allow the teachings of San Andres to improve the accuracy and reliability of the error detection protocol." *Id.* For at least the reasons stated above, Appellants respectfully assert that it would not be obvious to modify the teachings of San Andres with the step of *voting by all of the other nodes in the computer cluster* and therefore the Examiner has not presented a *prima facie* case of obviousness. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 9, 15 and 21 in view of the cited prior art.

San Andres does not teach or suggest "*applying the update to a local copy of the database* at each of the plurality of nodes in the computer cluster" as recited in claims 9 and 21 and similarly in claim 15. Instead, San Andres teaches that "the servers 120 that receive the update transaction from the Arbiter *respond by processing the update transaction*, and by *returning a status code that indicates the success or failure of the transaction.*" Column 19, Lines 25-28. San Andres further teaches that "*all replicated servers of the service group maintain local copies of the service's content data*, and provide user access to such data." Column 9, Lines 19-21. As interpreted by the Appellants, San Andres teaches that the replicated servers maintain local copies of the service's content data. Furthermore, as interpreted by the Appellants, San Andres teaches that the servers process the update transaction received from the Arbiter. However, as interpreted by the Appellants, San Andres *does not teach that the servers apply the update to a local copy of the database.* San

Andres simply teaches that the servers process the update transaction. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 9, 15 and 21 in view of the cited prior art.

San Andres does not teach or suggest that the "*node requesting update broadcasts results of update to all of the other nodes in the computer cluster*" as recited in claims 9 and 21 and similarly in claim 15. Instead, San Andres teaches that "when an application server of a service group receives a client request that indicates a modification to replicated service content data, the *server/service generates an update transaction and sends the update transaction to the Arbiter*. The Arbiter records the update transaction in a service-group specific transaction log ... and forwards the transaction for immediate processing to every application server in the server group ... The *application servers process the update transaction, and return status codes indicating, for each respective application server, the 'success' or 'failure' of the transaction.*" Column 3, Lines 26-38. As interpreted by the Appellants, San Andres teaches that the server receiving the request that indicates a modification generates an update transaction and sends the update transaction to the Arbiter. As interpreted by the Appellants, San Andres further teaches that the Arbiter forwards the update transaction to each server and that each server *returns a status code to the Arbiter* indicating the success or failure of the transaction. San Andres does not teach that the node requesting an update *broadcasts the results of the update to all of the other nodes* in the computer cluster. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 9, 15 and 21 in view of the cited prior art.

San Andres does not teach or suggest "*comparing, by all of the other nodes* in the computer cluster, the *update results to results of application of the update to the local copy of the database*" as recited in claims 9 and 21 and similarly in claim 15. Instead, San Andres teaches that "when one server 120 of the service group processes the dispatched transaction

differently than the other servers, *the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group*. In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent'* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance. San Andres does not teach that *all of the other nodes in the computer cluster compare* the update results to results of application of the update to the local copy of the database. Furthermore, San Andres does not teach that all of the other nodes in the computer cluster compare the *update results to results of application of the update to the local copy of the database.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 9, 15 and 21 in view of the cited prior art.

San Andres does not teach or suggest "*voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison*" as recited in claims 9 and 21 and similarly in claim 15. San Andres teaches that "whenever the Arbiter replicates a transaction, the Arbiter monitors the outcome of the transaction on each server 120 of the service group to ensure consistent processing of the transaction by all such servers. When *one or more servers 120 indicates a different outcome than the other servers of the service group, the Arbiter uses a voting scheme to resolve the conflict between the servers.*" Column 17, Lines 10-17. The Advisory Action and Office Action (dated September 25, 2001) states that the "Examiner is entitled to give claim limitations their broadest reasonable interpretation in light of the specification." Advisory Action, Page 2; Office Action (dated September 25, 2001), Page 4. Appellants respectfully point out to the Examiner that the Examiner is entitled to give claim limitations their broadest reasonable interpretation *in light*

of the specification. In particular, the *Examiner may not simply ignore language in the* claims. *All words in a claim must be considered in judging the patentability of that claim against the prior art. In re Wilson*, 424 F.2d 1382, 1385, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970); M.P.E.P. §2143.03. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance. San Andres does not teach *voting, by all of the other nodes in the computer cluster*. Furthermore, San Andres does not teach voting, by all of the other nodes in the computer cluster, *to approve update if a match results from the comparison*. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 9, 15 and 21 in view of the cited prior art.

San Andres does not teach or suggest "*voting, by any one of the other nodes* in the computer cluster, to *continue with update process if a match does not result from the comparison*" as recited in claim 10 and similarly in claim 16. Instead, San Andres teaches that "when one server 120 of the service group processes the dispatched transaction differently than the other servers, *the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group*. In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent'* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance*. San Andres does not teach that *voting by any of the other nodes in the computer cluster*. Furthermore, San Andres does not teach voting by any of the other nodes in the computer cluster *to continue with update process if a match does not result from the comparison*.

Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 10 and 16 in view of the cited prior art.

San Andres does not teach or suggest "*broadcasting an approval* of the *update to the database if all of the other nodes vote to approve the update*" as recited in claims 11, 22 and similarly in claim 17. Instead, San Andres teaches that "each update transaction replicated by the transaction replication service is stored in a transaction log. When a new application server is brought on-line, *previously-dispatched update transactions stored in the transaction log are dispatched in sequence to the new server* to bring the new server's content data up-to-date." Abstract. San Andres does not teach or suggest *broadcasting an approval* of the *update to the database*. Furthermore, San Andres does not teach or suggest broadcasting an approval of the update to the database *if all of the other nodes vote to approve the update*. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 11, 17 and 22 in view of the cited prior art.

San Andres does not teach or suggest "*if more than one of the plurality of nodes votes to continue*, performing a recovery process" as recited in claims 12, 23 and similarly in claim 18. Instead, San Andres teaches that "when one server 120 of the service group processes the dispatched transaction differently than the other servers, *the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group*. In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent'* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance*. As stated above, San Andres *does not teach voting by nodes*. Thus,

necessarily, San Andres does not teach *if more than one of the plurality of nodes votes to continue*, performing a recovery process. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 12, 18 and 23 in view of the cited prior art.

San Andres does not teach or suggest "*if more than a specified number of the nodes vote to continue, backing out the update to the database* as recited in claim 13, 24 and similarly in claim 19. Instead, San Andres teaches that "when one server 120 of the service group processes the dispatched transaction differently than the other servers, *the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group*. In the general case, the Arbiter uses a "majority rules" voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being "inconsistent"* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance*. As stated above, San Andres *does not teach voting by nodes*. Necessarily, San Andres does not teach *if more than a specified number of the nodes vote to continue, backing out the update to the database*. Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 13, 19 and 24 in view of the cited prior art.

San Andres does not teach or suggest "*if less than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes*" as recited in claims 14, 25 and similarly in claim 20. Instead, San Andres teaches that "when one server 120 of the service group processes the dispatched transaction differently than the other servers, *the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group*. In the general case, the Arbiter uses a 'majority rules'

voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent'* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance.* As stated above, San Andres *does not teach voting by nodes.* Necessarily, San Andres does not teach *if less than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claims 14, 20 and 25 in view of the cited prior art.

San Andres does not teach or suggest "*voting, by any one of the other nodes* in the computer cluster, to *continue with update process if a match does not result from the comparison*" as recited in claim 21. Instead, San Andres teaches that "when one server 120 of the service group processes the dispatched transaction differently than the other servers, *the Arbiter uses a voting scheme to decide which server or servers are to be taken off-line within the service group.* In the general case, the Arbiter uses a 'majority rules' voting scheme. Under the majority rules scheme, *if a minority number of servers 120 of the service group report a different outcome than the other servers, the minority servers are treated as being 'inconsistent'* with the final outcome, and are taken off-line." Column 19, Lines 46-55. As interpreted by the Appellants, San Andres teaches that an *Arbiter* may *use a voting scheme to decide which application servers are deemed to be consistent and take application servers that are inconsistent off-line for maintenance.* San Andres does not teach *voting by any of the other nodes in the computer cluster.* Furthermore, San Andres does not teach voting by any of the other nodes in the computer cluster *to continue with update process if a match does not result from the comparison.* Accordingly, one ordinarily skilled in the art would not be capable to re-create claim 21 in view of the cited prior art.

As a result of the foregoing, Appellants respectfully assert that the Examiner's *prima facie* case of obviousness is not taught or suggested by the cited prior art since there are numerous claim limitations, and thus one skilled in the art would not have been able to create the claimed invention in view of the cited prior art.

IX. CONCLUSION

For the reasons noted above, the rejection of claims 1-25 are in error. Appellants respectfully request reversal of the rejections and allowance of claims 1-25.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorney for Appellants

By: _____

Robert A. Voigt, Jr.
Reg. No. 47,159
Barry S. Newberger
Reg. No. 41,527
Kelly K. Kordzik
Reg. No. 36,571

5400 Renaissance Tower
1201 Elm Street
Dallas, Texas 75270-2199
(512) 370-2832

::ODMA\PCDOCS\Austin_1\186748\2
1162:7047-P280US

APPENDIX

1. A method for maintaining a consistent set of replicas of a database within a computer cluster, comprising the steps of:

each node in the computer cluster receiving a database update request;

each node in the computer cluster voting based on a functional outcome of the database update request; and

detecting an out-of-sync condition as a result of a different functional outcome.

2. The method as recited in claim 1, wherein the out-of-sync condition is an error.

3. The method as recited in claim 1, further comprising the step of:

refreshing the database in response to the detecting step.

4. The method as recited in claim 1, further comprising the step of:

resetting cluster membership in response to the detecting step.

5. The method as recited in claim 1, further comprising the step of:

blocking further participation by the node having the out-of-sync condition in response to the detecting step.

6. The method as recited in claim 1, further comprising the step of:

declaring an end-of-transaction state on update voting completion when the database update is being done in a transactional manner.

7. The method as recited in claim 6, further comprising the step of:

backing out an update when update voting does not meet a criteria established for success.

8. The method as recited in claim 7, wherein the criteria established for success is that no more than one node has inconsistent results.

9. A method for maintaining a consistent set of replicas of a database within a computer cluster, comprising the steps of:

broadcasting an update to a database shared among a plurality of nodes in the computer cluster;

applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster;

node requesting update broadcasts results of update to all of the other nodes in the computer cluster;

comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database; and

voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison.

10. The method as recited in claim 9, further comprising the step of:

voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison.

11. The method as recited in claim 9, further comprising the step of:

broadcasting an approval of the update to the database if all of the other nodes vote to approve the update.

12. The method as recited in claim 10, further comprising the step of:

if more than one of the plurality of nodes votes to continue, performing a recovery process.

13. The method as recited in claim 12, wherein the recovery process further comprises the step of:

if more than a specified number of the nodes voted to continue, backing out the update to the database.

14. The method as recited in claim 12, wherein the recovery process further comprises the step of:

if less than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes.

15. A computer cluster operable for maintaining a consistent set of replicas of a database within the computer cluster, comprising:

a group services client operable for broadcasting an update to a database shared among a plurality of nodes in the computer cluster;

the plurality of nodes coupled to the computer cluster operable for applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster;

circuitry for broadcasting results of the update to all of the other nodes in the computer cluster;

circuitry for comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database; and

circuitry for voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison.

16. The computer cluster as recited in claim 15, further comprising:

circuitry for voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison.

17. The computer cluster as recited in claim 15, further comprising:

circuitry for broadcasting an approval of the update to the database if all of the other nodes vote to approve the update.

18. The computer cluster as recited in claim 16, further comprising:

if more than one of the plurality of nodes votes to continue, circuitry for performing a recovery process.

19. The computer cluster as recited in claim 18, wherein the recovery process further comprises:

if more than a specified number of the nodes voted to continue, circuitry for backing out the update to the database.

20. The computer cluster as recited in claim 18, wherein the recovery process further comprises:

if less than a specified number of the nodes voted to continue, circuitry for performing the recovery process on the specified number of the nodes.

21. A computer program product adaptable for storage on a computer readable medium, the computer program product operable for maintaining a consistent set of replicas of a database within a computer cluster, comprising the program steps of:

broadcasting an update to a database shared among a plurality of nodes in the computer cluster;

applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster;

node requesting update broadcasts results of update to all of the other nodes in the computer cluster;

comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database;

voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison; and

voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison.

22. The computer program product as recited in claim 21, further comprising the program step of:

broadcasting an approval of the update to the database if all of the other nodes vote to approve the update.

23. The computer program product as recited in claim 22, further comprising the program step of:

if more than one of the plurality of nodes votes to continue, performing a recovery process.

24. The computer program product as recited in claim 23, wherein the recovery process further comprises the program step of:

if more than a specified number of the nodes voted to continue, backing out the update to the database.

25. The computer program product as recited in claim 24, wherein the recovery process further comprises the program step of:

if less than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes.

A method for maintaining a consistent set of replicas of a database within a computer cluster, comprising the steps of:

broadcasting an update to a database shared among a plurality of nodes in the computer cluster;

applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster;

node requesting update broadcasts results of update to all of the other nodes in the computer cluster;

comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database; and

voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison.